ON TIE-BREAKERS AND OTHER EXOTIC TOOLS FOR THE NEH ALGORITHM

(updated version, 22-02-10)

Ramon Companys & Imma Ribas
DOE – ETSEIB – UPC

Este es el primero de tres trabajos dedicados a procedimientos de resolución de los problemas *flowshop* con criterio minimización del *makespan*, basados en la clásica heurística de Nawaz, Enscore y Ham. Consideramos dos casos el $Fm|prmu|C_{max}$ y el $Fm|block|C_{max}$. En el presente trabajo analizamos los procedimientos de desempate en la intercalación del segundo paso del NEH. Basamos las conclusiones en los experimentos realizados tomando como base los 120 ejemplares de las colecciones de Taillard. Son suficientemente conocidos y ejercen cierta fascinación sobre los investigadores. En anexo recogemos resultados sobre otras colecciones, más numerosas, generadas siguiendo los principios establecidos por Taillard. Al no disponer de personal auxiliar para realizar los experimentos computacionales, dicho anexo se irá completando en la medida de lo posible.

El segundo trabajo estará dedicado a analizar el efecto de diferentes procedimientos de ordenación en el paso 1, algunos propuestos por otros autores, otros originales. El tercer trabajo se dedicará al efecto de los procedimientos de mejora de las soluciones obtenidas en el paso 1 y el paso 2.

De esta forma dispondremos de manera ordenada un material muy considerable que hemos ido elaborando durante los últimos años, algunas veces siguiendo senderos tortuosos a los que nos condujeron los comentarios de algunos revisores, supuestamente bien intencionados.

1. INTRODUCCIÖN.

El *flow shop scheduling problem* (FSP) consiste en encontrar una secuencia de *n* trabajos (que denominaremos genéricamente “piezas”) que se deben procesar en *m* estaciones de trabajo (que denominaremos genéricamente “máquinas”) con el fin de minimizar una o varias medidas de eficiencia. El tiempo de proceso de la operación de la pieza i en la máquina j es $p_{j,i}$, siendo $p_{j,i} \geq 0$, donde j $\in$ {1, 2, …, m} indica la máquina e i $\in$ {1, 2, …, n} la pieza. En la ruta de todas las piezas el orden en que aparecen las máquinas es esencialmente el mismo, lo que permite numerar las máquinas 1, 2, …, m de tal forma que para cualquier pieza la operación siguiente a una operación en la máquina j tenga lugar en una máquina k con k > j. Si a todo i y j, $p_{j.i} > 0$ (supuesto que suele hacerse habitualmente) k = j + 1. La medida de eficiencia más estudiada, que aquí también consideramos, es el *makespan* o máximo del instante de finalización de las piezas.

Consideramos dos variantes. En la primera existe espacio ilimitado de almacenamiento de las piezas entre dos máquinas consecutivas y el orden de las piezas se mantiene constante en todas las máquinas. Esta variante se denomina *permutation flow shop* *scheduling problem* (PFSP) y siguiendo la notación propuesta por Graham et al. (1979) se denota como Fm|prmu|C_{max}.

En la segunda variante no existe espacio de almacenamiento entre dos máquinas sucesivas, por lo que si la pieza i termina su operación en la máquina j, y la máquina en la que debe realizarse la siguiente operación, la j +1, está todavía ocupada en la pieza precedente, la pieza i debe permanecer en la máquina j bloqueándola. Esta variante se denomina *blocking flow shop scheduling problem* (BFSP) y siguiendo la notación propuesta por Graham et al. (1979) se denota como Fm|block|C_{max}.

1.1. Modelos y formalización.

Vamos a formalizar ambos problemas. En el instante cero, hay *n* trabajos que deben ser procesados, en el mismo orden, en *m* máquinas. Cada trabajo va de la máquina 1 a la *m*. El tiempo de proceso de cada operación es $p_{j,i}$, siendo $p_{i,j} > 0$. Se considera que los tiempos de preparación están incluidos en el tiempo de proceso. La función objetivo considerada es la minimización del *makespan* que equivale a la maximización el uso global de las máquinas.

Dada una permutación, σ, de las *n* piezas, [k] indica el trabajo que ocupa la posición *k* en la secuencia. Dado un programa factible asociado a una permutación, se define $e_{j.k}$ como el instante inicial en que la pieza que ocupa la posición *k* empieza a procesarse en la máquina *j* y $f_{j,k}$ como el instante de finalización de dicha operación. El problema Fm|prmu|C_{max} se puede formalizar como sigue:

$$e_{1,1} = 0 \quad ; \quad C_{max}(\sigma) = [\text{MIN}]\, f_{m,n} \qquad (0)$$

$$e_{j,k} \geq f_{j,k-1} \qquad j = 1, 2, \ldots, m \qquad k = 1, 2, \ldots, n \qquad (1)$$

$$e_{j,k} \geq f_{j-1,k} \qquad j = 1, 2, \ldots, m \qquad k = 1, 2, \ldots, n \qquad (2)$$

$$e_{j,k} + p_{j,[k]} \leq f_{j,k} \qquad j = 1, 2, \ldots, m \qquad k = 1, 2, \ldots, n \qquad (3)$$

siendo las condiciones de contorno:

$f_{j,0} = 0 \ \forall j;\ f_{0,k} = 0\ \forall k$
.

El programa es semi-activo (con lo que se garantiza que el valor de $f_{m,n}$ sea el mínimo asociado a la secuencia σ) si la ecuación (3) se escribe como $e_{j,k} + p_{j,[k]} = f_{j,k}$ y las ecuaciones (1) y (2) se compactan en $e_{jk} = \max\{f_{j,k-1};\ f_{j-1,k}\}$.

El PFSP con el criterio indicado consiste en buscar una permutación σ tal que minimice el valor $C_{max}(\sigma)$.

Cuando no existe espacio de almacenaje entre las máquinas, problema BFSP, es necesario incluir una inecuación adicional (4) en la modelización de la situación, lo que conduce a:

$e_{1,1} = 0$; $C_{max}(\sigma) = [\text{MIN}]\ f_{m,n}$ (0)

$e_{j,k} \geq f_{j,k-1}$ $j = 1, 2, \ldots, m$ $k = 1, 2, \ldots, n$ (1)

$e_{j,k} \geq f_{j-1,k}$ $j = 1, 2, \ldots, m$ $k = 1, 2, \ldots, n$ (2)

$e_{j,k} + p_{j,[k]} \leq f_{j,k}$ $j = 1, 2, \ldots, m$ $k = 1, 2, \ldots, n$ (3)

$f_{j,k} \geq f_{j+1,k-1}$ $j = 1, 2, \ldots, m$ $k = 1, 2, \ldots, n$ (4)

siendo las condiciones de contorno:

$f_{j,0} = 0\ \ \forall j$; $f_{0,k} = 0$ y $f_{m+1,k} = 0\ \forall k$

Realmente las inecuaciones (1) no son necesarias[1], ya que están dominadas por el trio (2), (3) y (4)

El programa es semi-activo si las ecuaciones (3) y (4) se compactan como (teniendo en cuenta la observación anterior):

$$f_{j,k} = \min\{f_{j-1,k} + p_{j,[k]}, f_{j+1,k-1}\} \quad (4')$$

En este caso no es preciso introducir explícitamente la expresión que indica la no existencia de espacio de almacenamiento intermedio:

$$f_{j-1,k} = e_{j,k} \qquad \text{para } j = 2, 3, \ldots, m \text{ y } k = 1, 2, \ldots, n$$

Consecuencia de todo lo anterior es que el problema $Fm|prmu|C_{max}$ puede verse como una relajación del problema $Fm|block|C_{max}$ y por tanto el min C_{max} del primero es una cota inferior del min C_{max} del segundo, cuando los ejemplares considerados tienen idénticos valores de $p_{j.i}$.

Si algún valor del tiempo de proceso es nulo, $p_{j,i} = 0$, las expresiones anteriores deben ser adecuadamente modificadas.

1.2. Reversibilidad.

Pasamos ahora a considerar la reversibilidad de los problemas $Fm|prmu|C_{max}$ y $Fm|block|C_{max}$. Dado un ejemplar I (definido por una tabla m×n de valores de los tiempos de proceso $p_{j,i}$), que llamamos ejemplar directo, se puede determinar el ejemplar inverso I', con unos tiempos de proceso $p'_{j,i}$ calculados según (5) :

$$p'_{j,i} = p_{m-j+1,i} \qquad j = 1, 2, ..., m \quad i = 1, 2, ..., n \qquad (5)$$

Para una permutación **σ,** el valor de $C_{max}(\sigma)$ en **I** es el mismo que el obtenido en **I'** para la permutación inversa **σ',** $C_{max}(\sigma')$. Por lo tanto, el C_{max} mínimo es el mismo para los ejemplares I e I', y las permutaciones óptimas asociadas a ambos ejemplares son inversas unas de otras. En consecuencia, no importa buscar el min C_{max} en el ejemplar **I** o el **I'**, hallado en uno se conoce en el otro. Ambos problemas considerados

[1] Como ponen de manifiesto Wang et al. (2010)

$Fm|prmu|C_{max}$ y $Fm|block|C_{max}$ son reversibles en el sentido indicado. En el fondo para dichos problemas el ejemplar directo y el ejemplar inverso son dos visiones o versiones distintas del mismo ejemplar.

Este hecho, por lo menos para el problema $Fm|prmu|C_{max}$, es conocido desde hace unos 50 años. Algunos autores, como Brown and Lomnicki (1966) y McMahon and Burton (1967), empleando procedimientos de resolución basados en el *branch-and-bound* llegaron a la conclusión experimental, que el ejemplar inverso algunas veces podía resolverse con más facilidad que el ejemplar directo. Nosotros, también experimentalmente, hemos observado, además, que en algunos casos al aplicar el procedimiento B&B sobre el ejemplar directo mejora las soluciones, mientras que al aplicarlo sobre el ejemplar inverso mejora las cotas (o recíprocamente). Esta observación nos llevó a proponer el algoritmo LOMPEN (Companys, 1999).

Pinedo (1995), a pesar de lo trivial del tema, formaliza la relación entre los ejemplares directos e inversos a través de unos lemas, siguiendo su costumbre.

Hemos observado también que la aplicación de ciertas heurísticas no lleva a la misma solución (permutaciones inversas una de la otra) en el ejemplar directo que en el ejemplar inverso, lo que sirve como herramienta para mejorar el rendimiento de dichas heurísticas.

1.2. Los dos pasos del algoritmo NEH.

Diferentes autores han observado que el procedimiento NEH tal como lo propusieron Nawaz, Enscore and Ham (1983), y a pesar de que dichos autores lo describieron en tres fases[2], puede considerarse que consta de dos pasos:

(1) creación de una secuencia inicial de las piezas,
(2) procedimiento de inserción iterativo de las piezas siguiendo el orden de la secuencia inicial obtenida en el paso 1.

En la versión original el primer paso consiste en ordenar las piezas en orden no creciente del tiempo de proceso total $P_i = \sum_{j=1}^{m} p_{j,i}$, es decir en el orden LPT (*larger processing time*)

El segundo paso procede como sigue. De acuerdo a la secuencia obtenida en el paso 1, se toman las dos primeras piezas y se programan de forma que minimicen el makespan parcial, como si el problema tuviera sólo dos piezas, probando las dos permutaciones posibles de ambas piezas. A continuación, para K = 3 hasta *n*, insertar el K-th trabajo en una de las K posible posiciones de la secuencia parcial con el fin de minimizar el C_{max} del problema $Fm|prmu|C_{max}$ o $Fm|block|C_{max}$, según el caso, con *K* trabajos.

Dada la eficiencia de este procedimiento, especialmente en el problema $Fm|prmu|C_{max}$ (para el que fue propuesto inicialmente) pero también en el $Fm|block|C_{max}$ (según Leisten, 1990) se han presentado diferentes variantes en la literatura, la mayoría basadas en la forma de ordenar las piezas en el paso 1.

[2] A partir de aquí eludimos la palabra fase para reservarla con un significado diferente.

Posibilidades de empate.

En la versión original no se indica ningún procedimiento de desempate, ni en el paso 1 ni en el paso 2. En ambos pasos son posibles situaciones no totalmente definidas; por ejemplo en el paso 1, utilizando la ordenación LPT, si dos piezas distintas h e i son tales que $P_h = P_i$. ¿Cuál de las dos tiene prioridad? ¿Debe situarse h antes que i o bien i antes que h? La respuesta no es trivial dado que el orden inicial definido en el paso 1 influye decisivamente en la calidad de la permutación hallada en el paso 2. En cualquier tipo de ordenación basada en un índice (con la de Nagano and Moccelin, 2002, la de Kalczynski and Kamburowsky, 2008, etc.) los empates son posibles, pero no hemos sido capaces de hallar en la literatura indicaciones de cómo hacerles frente de manera eficiente.

En el paso 2 se produce empate cuando dos posiciones diferentes proporcionan el mismo *makespan*, al intentar intercalar una pieza i en la secuencia parcial. Aquí se han realizado diferentes propuestas Kalczynski and Kamburowski (2008), Dong et al. (2008) además de la propuesta propia, citada en Companys and Mateo (2007)

1.3. Estructura del trabajo.

En el presente trabajo realizamos experimentos con las diferentes maneras de hacer frente a los empates en el algoritmo NEH (prácticamente centrados en los empates en el paso 2). Utilizamos como base experimental las colecciones de Taillard (1993). Utilizamos como indicador de la calidad de las soluciones dadas por una heurística el índice de discrepancia relativa porcentual:

$$I_{r,x} = \frac{Heu_{r,x} - Best_x}{Best_x} \times 100$$

donde $Best_x$ es el mejor valor conocido del makespan del ejemplar x, mientras que $Heu_{r,x}$ es el valor obtenido por la heurística r al ejecutarse sobre el ejemplar x. Los mejores valores conocidos se han tomado, para el caso $Fm|prmu|C_{max}$, de la página Web de Taillard (2005) y, para el caso $Fm|block|C_{max}$, de la correspondiente a Companys and Ribas (2010) con la corrección que se indica. La fecha asociada a los valores del documento es el 14 de enero de 2010. Periódicamente, cuando se dispone de mejores soluciones, halladas mediante los *shaker algorithms* utilizados, se incorporan al documento. A fecha 22-02-2010 se han introducido las siguientes modificaciones:

22-02-10			
500×20	111	36609	4
	112	36939	4
	113	36654	4
	114	36641	4
	115	36583	4
	116	36917	4
	117	36518	4
	118	36845	4
	119	36641	4
	120	36866	4

Con los valores de los índices calculamos valores medios asociados a una colección e incluso valores medios globales (aunque en puridad estos últimos puedan carecer de mucho significado).

Distinguimos como heurísticas distintas la aplicación de un procedimiento sobre el ejemplar directo, la aplicación sobre el ejemplar inverso, y la aplicación sobre ambos reteniendo la mejor solución (herramienta que hemos aplicado ampliamente).

2. **EMPATES EN EL PRIMER PASO**.

Si la regla para poner orden es sencilla, por ejemplo consiste en calcular un índice entero para cada pieza y ordenar comparando los valores de dicho índice, normalmente se producirán empates. Con la regla LPT suelen producirse. Por ejemplo en la colección TAIL0111 con n = 500 y m = 20, el numero de valores distintos para un ejemplar de $P_i = \sum_{j=1}^{m} p_{j,i}$ es potencialmente 500 (las posibilidades van desde 500×1 = 500 hasta 500×99 = 49.500, pero sólo existen 500 piezas en un ejemplar), pero ninguno de los 10 ejemplares de la colección llega a esta cifra; el número máximo de valores distintos es 328. En la *tabla 1* hemos resumido la situación.

ejemplar	valores	no rep.	rep. 2 vec.	rep. 3 vec.	rep. 4 vec.	rep. 5 vec.	rep. 6 vec.	rep. 7 vec.	rep. 8 vec
TAIL111	304	181	76	29	13	3	1	1	
TAIL112	314	187	85	29	9	4			
TAIL113	311	185	76	38	11	1			
TAIL114	317	194	79	31	11	1	1		
TAIL115	310	185	78	34	10	1	2		
TAIL116	312	183	84	34	9	1	1		
TAIL117	316	192	79	32	11	2			
TAIL118	307	178	78	39	11	1			
TAIL119	328	203	87	29	9				
TAIL120	317	191	84	31	8	2	1		

Tabla 1: Empates en P_i de los ejemplares de la colección de Taillard 500×20

En el ejemplar TAIL111 existen muchas secuencias de las 500 piezas compatibles con el orden LPT, concretamente:

$$7! \times 6! \times (5!)^3 \times (4!)^{13} \times (3!)^{29} \times (2!)^{76} \approx 1{,}53 \times 10^{76}$$

Es mucho suponer que todas ellas conducirán, después del paso 2, a una permutación de la misma calidad. ¿Cómo elegir la mejor, o por lo menos, una buena? A falta de regla específica de desempate actúa la regla implícita y el resultado depende del orden de aparición de las piezas, es decir, de su numeración inicial. Si las mismas piezas hubiesen sido descritas en un orden diferente conducirían a otra solución, lo que en principio no parece muy razonable (los resultados de un algoritmo deberían ser independientes del orden en que se comunican al mismo los diferentes objetos que manipula).

La forma más simple de romper estos empates es añadir criterios de ordenación adicionales, organizados jerárquicamente. ¿Cuáles pueden ser estos criterios?

- Utilizar el primer índice de trapecios: $S_{1,i} = \sum_{j=1}^{m} (m - j) \cdot p_{j,i}$. En caso de empate en P_i elegir la pieza con $S_{1,i}$ mayor,
- Utilizar el segundo índice de trapecios: $S_{2,i} = \sum_{j=1}^{m} (j - 1) \cdot p_{j,i}$. En caso de empate en P_i elegir la pieza con $S_{2,i}$ menor,
- Utilizar el tercer índice de trapecios, la *slope* de Palmer (1965), salvo el signo: $S_{3,i} = S_{1,i} - S_{2,i}$. En caso de empate en P_i elegir la pieza con $S_{3,i}$ mayor,
- Utilizar un índice derivado de trapecios $S_{4,i} = \min \{ S_{1,i} , S_{2,i}\}$. En caso de empate en P_i elegir la pieza con $S_{4,i}$ mayor. Este criterio coincide con la aplicación a segundo nivel del procedimiento de ordenación de Kalczynski and Kamburowski (2008),
- En caso de empate en P_i elegir la pieza con $p_{1,i}$ mayor,
- En caso de empate en P_i elegir la pieza candidata mediante un sorteo,
- …

Partimos de la base que es más interesante tener en las primeras posiciones de la secuencia las piezas más pesadas, de mayor carga. El criterio que habíamos utilizado hasta ahora (muy poco justificable salvo por la unicidad de resultados) es obtener previamente la secuencia según el algoritmo de trapecios, que es única, y considerar que las piezas se presentan para la ordenación LPT en dicho orden.

Ninguno de los criterios adicionales, ni la renumeración indicada, ha mostrado, en el caso *permutation*, ventaja respecto al orden inicial de las piezas en las colecciones de Taillard (aspecto bastante sorprendente) por lo que en lo que sigue no hemos utilizado ninguno de ellos salvo si se indica lo contrario[3]. En el caso *blocking* la situación está menos definida (ver comentario a la *tabla 6*), pero hemos adoptado idéntico criterio. Comentarios adicionales se encuentran en el Anexo IV.

3. **EMPATES EN EL SEGUNDO PASO**.

¿Se producen este tipo de empates? Implícitamente se supone que en la intercalación se trabaja de izquierda a derecha (posiciones 1, 2, …, K) y que en caso de empate se retiene la primera posición explorada. Ambos supuestos son totalmente arbitrarios y enraizados con nuestras costumbres occidentales. ¿Por qué no recorrer la secuencia parcial de derecha a izquierda? ¿Por qué no adoptar en caso de empate la posición hallada la última? De hecho en los casos más simples aplicar el procedimiento al ejemplar inverso equivale a aplicarlo al directo resolviendo los empates en la forma contraria a la habitual; y las soluciones obtenidas en el ejemplar directo no son idénticas

[3] En aquellos procedimientos de desempate que tienden a conducir a la misma solución en el directo que en el inverso puede ser conveniente perturbar la presentación de las piezas para la ordenación LPT, ya que a fin de cuenta los valores peores se descartan y en cambio se aceptan los mejores. Esto ha dado buen resultado en el procedimiento de desempate DHC al comunicar para el LPT del inverso las piezas en el orden obtenido después de la intercalación en el directo.

a las obtenidas en el inverso. En la *tabla 2* se han analizado los resultados obtenidos en la colección de Taillard para n = 50 y m = 20, considerando los datos como pertenecientes a ejemplares del problema $Fm|prmu|C_{max}$

ejemplar	directo	inverso	dir+inv	best
TAIL051	4082	4006	4006	3850
TAIL052	3921	3958	3921	3704
TAIL053	3927	3866	3866	3640
TAIL054	3969	3953	3953	3720
TAIL055	3835	3872	3835	3610
TAIL056	3914	3861	3861	3681
TAIL057	3952	3927	3927	3704
TAIL058	3938	3914	3914	3691
TAIL059	3952	3970	3952	3743
TAIL060	4079	4036	4036	3756

Tabla 2: Valor de las permutaciones obtenidas aplicando el procedimiento NEH a la versión directa e inversa de los ejemplares de Taillard 50×20 (caso $Fm/prmu/C_{max}$).

En las columnas "directo" e "inverso" aparecen los valores de las permutaciones obtenidas aplicando el algoritmo NEH original el ejemplar directo o al ejemplar inverso. En la columna "dir+inv" el mejor de ambos valores y en la columna "best" el mejor valor conocido, que permite calcular el índice (*tabla 3*).

ejemplar	directo	inverso	dir+inv
TAIL051	6,025974	4,051948	4,051948
TAIL052	5,858531	6,857451	5,858531
TAIL053	7,884615	6,208791	6,208791
TAIL054	6,693548	6,263441	6,263441
TAIL055	6,232687	7,257618	6,232687
TAIL056	6,329802	4,889976	4,889976
TAIL057	6,695464	6,020518	6,020518
TAIL058	6,691953	6,041723	6,041723
TAIL059	5,583756	6,064654	5,583756
TAIL060	8,599574	7,454739	7,454739
media	6,659591	6,111086	5,860611

Tabla 3: Valor del índice de discrepancia relativa de las permutaciones obtenidas aplicando el procedimiento NEH a la versión directa e inversa de los ejemplares de Taillard 50×20 (caso $Fm/prmu/C_{max}$).

Pueden observarse notables diferencias entre los valores C_{max} del directo y del inverso, atribuibles en este caso a la forma de resolver los empates. Como en algunos casos es mejor el valor en el directo y en otros en el inverso se obtiene ventaja calculando ambos y reteniendo el mejor de los resultados (columna dir+inv). El valor medio del índice de discrepancia relativa mejora un 13,6 % respecto al directo y un 4,3 % respecto al inverso.

Unos resultados más globales (también asociados al caso Fm|prmu|C_{max}) teniendo en cuenta todas las colecciones son los de la *tabla 4*. En la ordenación LPT se consideran las piezas en el orden de aparición (i = 1, 2, …, n), en la intercalación se visitan las posibles posiciones de izquierda a derecha (posiciones 1, 2, …, K)

CR_1: solución NEH, ejemplar directo, los empates (en la ordenación inicial y en la inserción) se resuelven a favor del primer candidato,
CR_2: solución NEH, ejemplar directo, los empates (en la ordenación inicial y en la inserción) se resuelven a favor del último candidato,
CR_3: solución NEH, ejemplar inverso, los empates (en la ordenación inicial y en la inserción) se resuelven a favor del primer candidato,
CR_4: solución NEH, ejemplar inverso, los empates (en la ordenación inicial y en la inserción) se resuelven a favor del último candidato,
CR_5: solución NEH, mejor de directo e inverso, los empates (en la ordenación inicial y en la inserción) se resuelven a favor del primer candidato,
CR_6: solución NEH, mejor de directo e inverso, los empates (en la ordenación inicial y en la inserción) se resuelven a favor del último candidato,
CR_7: solución NEH, mejor de las cuatro posibilidades.

colección		CR_1	CR_2	CR_3	CR_4	CR_5	CR_6	CR_7
TA0001	20×5	3.300	2.916	2.817	3.348	2.492	2.622	2.492
TA0011	20×10	4.601	4,561	4.589	5.024	4.174	4.371	4.175
TA0021	20×20	3.731	3.686	3.609	3.746	3.360	3.437	3.360
TA0031	50×5	0.727	0.787	1.090	0.878	0.581	0.678	0.474
TA0041	50×10	5.073	5.127	5.697	5.307	4.973	4.819	4.610
TA0051	50×20	6.660	6.440	6.111	6.156	5.861	5.831	5.726
TA0061	100×5	0.527	0.464	0.496	0.457	0.378	0.339	0.331
TA0071	100×10	2.215	2.203	2.223	2.419	2.016	2.043	1.900
TA0081	100×20	5.344	5.388	5.617	5.682	5.196	5.283	4.973
TA0091	200×10	1,258	1.375	1.246	1.319	1.148	1.176	1.114
TA0101	200×20	4.408	4.379	4.566	4.451	4.196	4.196	3.999
TA0111	500×20	2.059	2.119	2.246	2,160	1.985	2.053	1.965
ALL		3.325	3.287	3.359	3.412	3.030	3.071	2.927

Tabla 4: Valores del índice de discrepancia medio de las permutaciones obtenidas aplicando el algoritmo NEH, en diversos supuestos, a los ejemplares de Taillard (caso Fm/prmu/C_{max}).

La *tabla 4* permite percibir, además del efecto significativo en la solución de la forma de resolver los empates (por ejemplo comparando CR_5 con CR_6), la mejora potencial al aplicar el mismo algoritmo a los ejemplares directo e inverso, reteniendo la mejor solución (comparando CR_5 con CR_1 o CR_6 con CR_2). En todo caso parece que es mayor el impacto de esta última.

¿Ocurre lo mismo en el caso Fm|block|C_{max}? La *tabla 5* es la equivalente a la *tabla 4* para dicho caso. Los efectos son semejantes y persiste la conveniencia de calcular los valores para las variantes directo e inverso de un ejemplar.

colección		CR_1	CR_2	CR_3	CR_4	CR_5	CR_6	CR_7
TA0001	20×5	5.580	5.316	5.249	5.342	4.894	4.903	4.766
TA0011	20×10	5.330	5.543	5.524	5.350	5.219	5.239	5.219
TA0021	20×20	3.464	3.423	3.480	3.391	3.365	3.256	3.248
TA0031	50×5	8.593	8.572	8.598	8.672	7.746	8.084	7.598
TA0041	50×10	7.787	7.710	7.843	7.897	7.522	7.501	7.195
TA0051	50×20	7.331	7.255	7.219	7.409	6.846	6.927	6.785
TA0061	100×5	8.342	8.479	8.204	8.663	7.924	8.020	7.643
TA0071	100×10	8.039	7.490	7.664	7.633	7.517	7.190	6.933
TA0081	100×20	5.616	6.203	6.179	5.739	5.516	5.675	5.303
TA0091	200×10	7.825	7.716	7.818	7.869	7.608	7.569	7.394
TA0101	200×20	5.654	5.460	4.491	5.465	5.244	5.134	4.995
TA0111	500×20	4.371	4.738	4.695	4.472	4.317	4.411	4.256
ALL		6.495	6.492	6.497	6.492	6.143	6.159	5.945

Tabla 5: Valores del índice de discrepancia medio de las permutaciones obtenidas aplicando el algoritmo NEH, en diversos supuestos, a los ejemplares de Taillard (caso $Fm/block/C_{max}$).

En este caso, como ya hemos dicho, hemos detectado que la renumeración de las piezas antes de LPT puede mejorar los resultados. Utilizando el orden de trapecios en el ejemplar directo para reordenar las piezas los resultados obtenidos pueden observarse en la *tabla 6* que debe compararse con la *tabla 5*: No hemos reordenado de acuerdo al orden trapecios en el ejemplar inverso ya que conduciría a los mismos valores alternados. Los valores medios de los índices son ligeramente mejores, pero no hemos aplicado esta reordenación en lo que sigue. Ver los comentarios adicionales del Anexo IV.

colección		CR_1	CR_2	CR_3	CR_4	CR_5	CR_6	CR_7
TA0001	20×5	5.291	5.344	5.222	5.631	4.852	4.945	4.766
TA0011	20×10	5.331	5.543	5.524	3.350	5.219	5.239	5.219
TA0021	20×20	3.464	3.423	3.480	3.391	3.365	3.256	3.248
TA0031	50×5	8.304	8.505	8.990	8.796	7.828	8.193	7.691
TA0041	50×10	8.114	7.678	8.158	7.733	7.940	7.378	7.299
TA0051	50×20	7.387	7.338	7.230	7.404	6,844	6.989	6.774
TA0061	100×5	8.345	8.194	8.113	8.481	7.661	7.916	7.505
TA0071	100×10	7.694	7.811	7.374	7.723	7.223	7.442	7.107
TA0081	100×20	5.926	6.240	6.117	5.468	5.727	5.401	5.202
TA0091	200×10	7.870	7.711	7.570	7.714	7.409	7.520	7.309
TA0101	200×20	5.437	5.478	5.587	5.159	5.246	5.032	4.934
TA0111	500×20	4.438	4.574	4.503	4.5622	4.320	4.427	4.250
ALL		6.467	6.487	6.489	6.451	6.136	6.145	5.942

Tabla 6: Valores del índice de discrepancia medio de las permutaciones obtenidas aplicando el algoritmo NEH, en diversos supuestos, con reordenación previa a los ejemplares de Taillard (caso $Fm/block/C_{max}$).

También es notable observar que las discrepancias en el caso Fm|block|C_{max} son del orden del doble de las del caso Fm|prmu|C_{max}. Si tenemos en cuenta que los valores *Best$_x$* utilizados están obviamente más ajustados en el segundo que en el primero, solo cabe concluir la superior eficiencia del algoritmo NEH cuando existen almacenamientos intermedios ilimitados entre las máquinas.

4. PROCEDIMIENTOS PROPUESTOS PARA RESOLVER LOS EMPATES EN EL SEGUNDO PASO.

Se han considerado cuatro procedimientos de desempate:

TM1: tiempo muerto, versión clásica, (Companys and Mateo, 2004)
TM2: tiempo muerto, versión renovada Companys and Ribas, 2009),
KK: propuesta por Kalczynsky and Kamburowski (2007)
DHC: propuesta por Dong et al. (2008)

A continuación se describen los cuatro y su adaptación al caso Fm|block|C_{max} (ya que fueron diseñados para el caso Fm|prmu|C_{max})

4.1. Tiempo muerto, versión clásica (TM1).

En caso de empate en el valor C_{max} de dos posibles posiciones de inserción se elige aquella con mejor valor del tiempo muerto total TIT = $\sum_{j=1}^{m} IT(j)$ donde

$$IT(j) = f_{j,} - \sum_{k=1}^{K} p_{j,[k]} \ .$$

K es el número de piezas que componen la secuencia parcial (contando la pieza a insertar). Las fórmulas anteriores se utilizan tanto en el PFSP como en el BFSP, por lo que en éste último en el cómputo se incluye además del tiempo muerto de las máquinas el tiempo de bloqueo.

4.2. Tiempo muerto, versión renovada (TM2).

Es similar al anterior, pero se prescinde del *front idle time*, en la expresión de IT(j)

$$IT(j) = f_{j,K} - e_{j,1} - \sum_{k=1}^{K} p_{j,[k]}$$

Las fórmulas anteriores se utilizan tanto en el PFSP como en el BFSP.

Tanto en este caso como en el anterior $\sum_{j=1}^{m}\sum_{k=1}^{K} p_{j,[k]}$ es constante para todas las posiciones que están siendo contrastadas, por lo que el término correspondiente puede omitirse en las comparaciones.

4.3. Procedimiento KK.

Utiliza dos índices a_i y b_i, cuyo valor menor $c_i = \min \{ a_i , b_i \}$ también sirve a los autores para la ordenación en el primer paso

$$a_i = \sum_{j=1}^{m} (\frac{(m-1)\cdot(m-2)}{2} + m - j)\cdot p_{j,i} = \frac{(m-1)\cdot(m-2)}{2}\cdot P_i + S_{1,i}$$

$$b_i = \sum_{j=1}^{m} (\frac{(m-1)\cdot(m-2)}{2} + j - 1)\cdot p_{j,i} = \frac{(m-1)\cdot(m-2)}{2}\cdot P_i + S_{2,i}$$

$$c_i = \frac{(m-1)\cdot(m-2)}{2}\cdot P_i + S_{4,i}$$

En los empates en la intercalación:

Si $a_i \leq b_i$ entonces la posición elegida es la más cercana a la cabeza

Si $a_i > b_i$ entonces la posición elegida es la más cercana a la cola

que es equivalente a

Si $S_{3,i} \leq 0$ entonces la posición elegida es la más cercana a la cabeza

Si $S_{3,i} > 0$ entonces la posición elegida es la más cercana a la cola

(la dualidad de los signos $\leq$ y $>$ cubre todas las posibilidades pero no es enteramente coherente)

Las expresiones anteriores se utilizan tanto en el PFSP como en el BFSP.

4.4. Procedimiento DHC

Si C_{max} parcial es el mismo al insertar la pieza i en varias posiciones, para cada una de ellas t sea $e_{j,t}$ y $f_{j,t}$ los instantes más tempranos en que i puede empezar y terminar su operación en la máquina j, y $e'_{j,t}$ y $f'_{j,t}$ los instantes más tardíos compatibles con el valor C_{max} (calculables a partir de los valores de las "matrices" de Taillard). Guardando las distancias (ya que el camino crítico puede estar constituido por varias operaciones sucesivas de la pieza i) $f'_{j,t} - e_{j,t} - p_{j,i}$ viene a ser el margen total en la máquina j si la pieza i se inserta en la posición t.

Los autores calculan dos valores asociados a la posición t (para la pieza i)

$$E_t = \frac{1}{m}\cdot\sum_{j=1}^{m}\frac{p_{j,i}}{f'_{j,t} - e_{j,t}} \; ; \; D_t = \sum_{j=1}^{m}\left(\frac{p_{j,i}}{f'_{j,t} - e_{j,t}} - E_t\right)^2 \quad \text{para } t = 1, 2, \ldots, k$$

Obviamente en la posición última k, $f'_{j,k}$ es igual a la fecha más tardía de terminación de la pieza que ocupa la posición t; $e_{j,1}$ es igual a la fecha mínima de comienzo de la primera pieza en la máquina j. Dicen los autores que si $f'_{j,t} = e_{j,t}$ entonces $p_{j,i}$ debe ser igual a cero y el cociente se iguala a cero (cosa que no puede ocurrir si todo $p_{j,i} > 0$).

En caso de empates se elige la posición t con el menor valor D_t.

Dada la interpretación como margen total, los cocientes de los que E_t es la media deben ser menores o iguales a uno, y marcan, de alguna manera, el "colchón" disponible (aunque sea a la inversa).

Debido a las relaciones de geometría de masas ampliamente conocidas, nosotros preferimos calcular $\sum_{j=1}^{m} \frac{p_{j,i}}{f'_{j,t} - e_{j,t}}$ y $\sum_{j=1}^{m} \left(\frac{p_{j,i}}{f'_{j,t} - e_{j,t}} \right)^2$, pasando posteriormente a calcular D_t.

En el BFSP se utilizan los mismos conceptos que en el PFSP, determinando adecuadamente $e_{j,t}$ y $f'_{j,t}$.

4.4. Experiencia computacional en el problema $Fm|prmu|C_{max}$

Empezamos con los cuatro procedimientos elementales descritos. Los valores de la *tabla 6* corresponden a la aplicación del procedimiento a partir de la ordenación inicial LPT (con los empates resueltos a favor de la pieza de menor índice) a los ejemplares directo e inverso, reteniendo el mejor valor obtenido (la columna NEH0 corresponde a la ausencia de desempate, es decir la resolución a favor del primer candidato, y es idéntica a la columna CR_5 de la *tabla 7* ya que ésta es coherente con la ausencia de preordenación de las piezas antes de LPT). Puede observarse que globalmente los mejores resultados los proporciona TM1, seguido de DCH. Curiosamente TM2 y KK proporcionan perores resultados globales que la ausencia de regla de desempate. Obviamente TM1 domina TM2. También resulta obvio que la dimensión de los ejemplares influye en la eficiencia de los procedimientos.

		NEH0	TM1	TM2	KK	DHC
TA0001	20×5	2.492	**2.130**	2.239	2.729	2.483
TA0011	20×10	4.174	4.107	**3.811**	4.312	4.126
TA0021	20×20	**3.360**	3.495	3.477	3.407	3.703
TA0031	50×5	0.581	0.541	**0.535**	0.588	0.713
TA0041	50×10	4.973	**4.387**	4.529	4.875	4.803
TA0051	50×20	5.861	**5.671**	5.834	6.424	6.254
TA0061	100×5	0.378	0.381	**0.267**	0.397	0.446
TA0071	100×10	2.016	1.768	**1.714**	1.771	1.958
TA0081	100×20	5.196	4.904	**4.826**	5.284	5.014
TA0091	200×10	1.148	1.053	0.949	1.127	**0.913**
TA0101	200×20	4.196	**3.849**	3.974	4.232	3.852
TA0111	500×20	1.985	1.819	1.860	2.030	**1.682**
todos		3.030	2.842	**2.835**	3.098	2.996

Tabla 7: Aplicación del algoritmo NEH a los ejemplares de Taillard (caso $Fm/prmu/C_{max}$) utilizando diversas reglas de desempate puras.

Habida cuenta de la observación al pie de la página 6, hemos repetido la aplicación de los procedimientos con desempate KK y DCH (en los que la diferenciación entre directo e inverso es muy poco acentuada) utilizando como orden previo antes de LPT para el inverso el orden resultante de la intercalación en el directo ("orden perturbado"), en otras palabras la secuencia LPT no es la misma en el directo que en el inverso. Hemos obtenido los valores de las columnas KKp y DCHp de la *tabla 6bis*. Se ha producido suficiente mejora como para superar, en el caso KKp. la ausencia de empate, pero no la suficiente como para superar a TM1.

		NEH0	TM1	TM2	KKp	DHCp
TA0001	20×5	2.492	**2.130**	2.239	2.570	2.483
TA0011	20×10	4.174	4.107	**3.811**	4.312	4.126
TA0021	20×20	**3.360**	3.495	3.477	3.376	3.703
TA0031	50×5	0.581	0.541	**0.535**	0.588	0.601
TA0041	50×10	4.973	**4.387**	4.529	4.603	4.678
TA0051	50×20	5.861	**5.671**	5.834	6.181	5.960
TA0061	100×5	0.378	0.381	**0.267**	0.397	0.415
TA0071	100×10	2.016	1.768	**1.714**	1.641	1.570
TA0081	100×20	5.196	4.904	4.826	5.054	**4.667**
TA0091	200×10	1.148	1.053	0.949	1.096	**0.904**
TA0101	200×20	4.196	3.849	3.974	4.033	**3.690**
TA0111	500×20	1.985	1.819	1.860	1.882	**1.619**
todos		3.030	2.842	**2.835**	2.978	2.868

Tabla 7bis: Aplicación del algoritmo NEH a los ejemplares de Taillard (caso $Fm/prmu/C_{max}$) utilizando diversas reglas de desempate puras y preordenando el inverso.

Las reglas de los procedimientos TM1 y TM2 son susceptibles, a su vez, de producir empates ya que en ambos casos TIT es un número entero, por ello puede ser conveniente asociarles una segunda regla. La regla de KK es finalista, y por tanto no sometida a empates, mientras que la regla DHC aunque teóricamente los puede sufrir, al ser D_t un valor racional no negativo y no superior a uno, ello es altamente improbable, a menos de establecer un umbral tal que si la diferencia de dos valores está dentro del mismo se consideren dichos valores idénticos a los efectos del empate. Por ello los procedimientos analizados en *la tabla 8* son TM2+KK y TM2+DHC.

		NEH0	TM1	TM1+ KK	TM1+ DHC	TM2	TM2+ KK	TM2+ DHC
TA0001	20×5	2.492	**2.130**	2.334	2.224	2.239	2.284	2.193
TA0011	20×10	4.174	4.107	3.937	3.930	**3.811**	3.979	3.991
TA0021	20×20	3.360	3.495	**3.324**	3.441	3.477	3.463	3.441
TA0031	50×5	0.581	0.541	0.587	**0.441**	0.535	0.517	0.506
TA0041	50×10	4.973	4.387	**4.003**	4.285	4.529	4.297	4.586
TA0051	50×20	5.861	5.671	5.638	5.503	5.834	5.794	**5.452**
TA0061	100×5	0.378	0.381	0.406	0.398	**0.267**	0.414	0.396
TA0071	100×10	2.016	1.768	1.861	**1.431**	1.714	1.848	1.560
TA0081	100×20	5.196	4.904	4.936	**4.807**	4.826	5.049	4.898
TA0091	200×10	1.148	1.053	0.822	0.851	0.949	0.924	**0.797**
TA0101	200×20	4.196	3.849	3.806	3.675	3.974	3.849	**3.656**
TA0111	500×20	1.985	1.819	1.824	**1.660**	1.860	1.801	1.727
todos		3.030	2.842	2.790	**2.721**	2.835	2.851	2.767

Tabla 8: Aplicación del algoritmo NEH a los ejemplares de Taillard (caso $Fm/prmu/C_{max}$) utilizando diversas reglas de desempate mixtas.

Las reglas adicionales mejoran los resultados en el caso de TM1, pero, aparentemente, la regla KK como adicional a TM2 los estropea; en cambio la regla DHC, en las mismas circunstancias, los mejora. En conclusión, vistos los resultados, para el problema $Fm|prmu|C_{max}$ el método de desempate recomendado es TM1+DHC (salvo si consideraciones de consumo de tiempo aconsejan suprimir la regla adicional y limitarse a TM1).

4.5. Experiencia computacional en el problema $Fm|block|C_{max}$

Empezamos otra vez con los cuatro procedimientos elementales. Los valores de la *tabla 9* corresponden a la aplicación del procedimiento a partir de la ordenación inicial LPT (con los empates resueltos a favor de la pieza de menor índice) a los ejemplares directo e inverso, reteniendo el mejor valor obtenido (la columna NEH0 corresponde a la ausencia de desempate, y es idéntica a la columna CR_5 de la *tabla 5*). Puede observarse que globalmente los mejores resultados los proporciona TM2, y que es el único procedimiento que mejora (por muy poco) la ausencia de regla. Obviamente TM1 domina TM2.

		NEH0	TM1	TM2	KK	DHC
TA0001	20×5	4.894	5.018	5.164	5.243	4.989
TA0011	20×10	5.219	5.300	5.264	5.520	**5.194**
TA0021	20×20	3.365	3.386	3.386	3.395	**3.290**
TA0031	50×5	7.746	7.867	7.951	8.143	8.212
TA0041	50×10	7.522	7.300	**7.295**	7.373	7.562
TA0051	50×20	6.846	7.161	7.136	7.173	7.085
TA0061	100×5	7.924	8.143	**7.855**	8.598	8.004
TA0071	100×10	7.517	7.325	**7.273**	7.805	7.342
TA0081	100×20	5.516	5.668	5.662	6.083	5.850
TA0091	200×10	7.608	**7.279**	7.310	7.472	7.522
TA0101	200×20	5.244	5.076	**5.052**	5.201	5.150
TA0111	500×20	4.317	**4.289**	4.296	4.454	4.430
todos		6.143	6.151	**6.137**	6.373	6.219

Tabla 9: Aplicación del algoritmo NEH a los ejemplares de Taillard (caso $Fm/block/C_{max}$) utilizando diversas reglas de desempate puras.

Dado que la diferencia entre directo e inverso es muy poca con la regla KK, y algo más importante con la DCH, procedemos a los cálculos de KKp y DCHp (*tabla 7bis*) en la que vemos que la mejora obtenida no es suficiente para superar TM2.

		NEH0	TM1	TM2	KKp	DHCp
TA0001	20×5	4.894	5.018	5.164	5.071	**4.876**
TA0011	20×10	5.219	5.300	5.264	5.520	**5.194**
TA0021	20×20	3.365	3.386	3.386	3.374	**3.269**
TA0031	50×5	7.746	7.867	7.951	7.927	8.184
TA0041	50×10	7.522	7.300	**7.295**	7.246	7.554
TA0051	50×20	6.846	7.161	7.136	6.966	7.144
TA0061	100×5	7.924	8.143	**7.855**	8.346	7.899
TA0071	100×10	7.517	7.325	**7.273**	7.201	7.430
TA0081	100×20	5.516	5.668	5.662	5.552	5.782
TA0091	200×10	7.608	**7.279**	7.310	7.348	7.630
TA0101	200×20	5.244	5.076	**5.052**	5.028	5.136
TA0111	500×20	4.317	**4.289**	4.296	4.330	4.376
todos		6.143	6.151	**6.137**	6.159	6.206

Tabla 9bis: Aplicación del algoritmo NEH a los ejemplares de Taillard (caso $Fm/block/C_{max}$) utilizando diversas reglas de desempate puras y preordenando el inverso.

Probamos las mismas combinaciones que antes (pero tomando como base TM2 que parece poseer cierta ventaja respecto TM1):

		NEH0	TM2	TM2+KK	TM2+DHC
TA0001	20×5	4.894	5.164	5.221	**4.894**
TA0011	20×10	5.219	5.264	5.226	**5.219**
TA0021	20×20	3.365	3.386	3.386	**3.269**
TA0031	50×5	7.746	7.951	8.261	8.082
TA0041	50×10	7.522	**7.295**	**7.254**	7.412
TA0051	50×20	6.846	7.136	6.946	6.970
TA0061	100×5	7.924	**7.855**	**7.756**	8.088
TA0071	100×10	7.517	**7.273**	7.434	7.348
TA0081	100×20	5.516	5.662	5.728	5.657
TA0091	200×10	7.608	7.310	7.629	7.487
TA0101	200×20	5.244	**5.052**	5.441	5.294
TA0111	500×20	4.317	**4.296**	4.436	4.328
todos		6.143	**6.137**	6.267	6.171

Tabla 10: Aplicación del algoritmo NEH a los ejemplares de Taillard (caso Fm /block /C_{max}) utilizando diversas reglas de desempate mixtas.

En este caso los resultados no son muy alentadores, lo que nos lleva a recomendar para el problema Fm | block | C_{max} la regla TM2 o la ausencia de regla de desempate.

4.6. Tiempos de proceso

Los experimentos se realizaron en un PC Intel Core 2 Duo E8400 CPU, 3GHz y 2GB de memoria RAM. La codificación se realizó en BC7. Los valores de los tiempos no son muy significativos debido al proceso de discretización en la medida de intervalos, la intervención de numerosos elementos aleaforios y no haber repetido varias veces el mismo cálculo para disponer de una valor medio significativo. No obstante son muy indicativos de la variación de dicho tiempo al aumentar la complejidad de los ejemplares. Sólo indico valores para los casos más significativos desde el punto de vista de la calidad de los resultados.

		NEH0	TM1	TM1+ KK	TM1+ DHC	TM2	TM2+ KK	TM2+ DHC
TA0001	20×5	0.000	0.005	0.000	0.000	0.006	0.000	0.006
TA0011	20×10	0.006	0.000	0.000	0.000	0.000	0.000	0.000
TA0021	20×20	0.005	0.000	0.006	0.006	0.005	0.000	0.005
TA0031	50×5	0.005	0.005	0.011	0.005	0.005	0.011	0.011
TA0041	50×10	0.016	0.011	0.022	0.033	0.016	0.017	0.033
TA0051	50×20	0.016	0.039	0.022	0.038	0.034	0.038	0.066
TA0061	100×5	0.049	0.055	0.061	0.099	0.033	0.060	0.049
TA0071	100×10	0.055	0.060	0.121	0.104	0.127	0.065	0.203
TA0081	100×20	0.242	0.263	0.137	0.423	0.137	0.264	0.220
TA0091	200×10	0.418	0.423	0.846	0.654	0.851	0.434	1.258
TA0101	200×20	1.719	1.757	1.801	2.873	0.918	1.813	2.226
TA0111	500×20	12.864	12.709	13.061	17.642	13.105	13.166	17.680

Tabla 11: Tiempo CPU en segundos por ejemplar al aplicar el procedimiento a los ejemplares en el supuesto Fm /prmu /C_{max}.

		NEH0	TM2	TM2+KK	TM2+DHC
TA0001	20×5	0.000	0.000	0.000	0.000
TA0011	20×10	0.000	0.000	0.000	0.000
TA0021	20×20	0.000	0.000	0.006	0.005
TA0031	50×5	0.005	0.006	0.006	0.016
TA0041	50×10	0.022	0.011	0.011	0.027
TA0051	50×20	0.017	0.034	0.038	0.082
TA0061	100×5	0.077	0.049	0.049	0.072
TA0071	100×10	0.082	0.165	0.165	0.280
TA0081	100×20	0.279	0.165	0.307	0.274
TA0091	200×10	0.648	1.291	0.659	1.938
TA0101	200×20	2.136	2.170	2.142	2.873
TA0111	500×20	713.367	838.756	704.599	1043.805

Tabla 12: tiempo CPU en segundos por ejemplar al aplicar el procedimiento a los ejemplares en el supuesto $Fm|block|C_{max}$.

En el caso $Fm|block|C_{max}$ existe una falta de alineación entre los valores de la colección TA0111 y el resto. Ello es debido a que los valores de C_{max} de dichos ejemplares obliga a la utilización de la doble longitud en algunos enteros, lo que ralentiza los cálculos.

5. CONCLUSIONES

Hemos mostrado que en los algoritmos basados en la heurística NEH son posibles, a diferentes niveles, la aparición de situaciones ambiguas debidas a empates, cuya resolución no es trivial. La herramienta que propugnamos, aplicar el mismo algoritmo al ejemplar directo y al ejemplar inverso reteniendo la mejor solución, no es más, en el fondo, que aprovechar eficientemente las diferencias ocasionadas por la resolución alternativa de los empates.

Utilizando como base experimental las colecciones de Taillard hemos deducido que para el caso *permutation* puede ser conveniente utilizar como herramienta para el desempate en la intercalación la aplicación de las reglas TM1+DHC, o, en su defecto, de la TM1 únicamente. Análogamente para el caso *blocking* les herramientas correspondientes son TM2 o ausencia de regla (regla FIFO). La utilización de estas conclusiones en todos los problemas $Fm|prmu|C_{max}$ y el $Fm|block|C_{max}$ parece presuponer que consideramos las colecciones de Taillard como el substrato del universo de los problemas *flowshop*. Ello no es así, no obstante las consideramos suficientemente representativas, y nuestros experimentos nos muestran que nada se opone a suponer que las reglas elegidas funcionarán bien en otros casos.

Una vez bien definidas las reglas de desempate el procedimiento de intercalación es un mecanismo determinista tal que dada una secuencia σ devuelve otra secuencia $\hat{\sigma}$. Podemos representar dicho mecanismo mediante la letra griega Φ, de forma que podemos escribir simbólicamente $\hat{\sigma} = \Phi(\sigma)$. Posiblemente diferentes σ puedan conducir a la misma $\hat{\sigma}$. Nos interesa una evaluación asociada a $\hat{\sigma}$: $C_{max} = f(\hat{\sigma})$. Como diferentes σ conducen a diferentes $\hat{\sigma}$, con evaluaciones también diferentes, la pregunta que podemos hacernos es: ¿cuál es la mejor secuencia σ para nuestros fines? (no es seguro que en el conjunto de todas las $\hat{\sigma}$ figure alguna solución óptima). Framinan et al. (2.003) examinan 176 reglas para generar una σ, sorprendentemente no se preocupan de la manera de resolver los empates, ni los que se producen en la formación de σ a partir de la regla ni los que se producen en la intercalación.

Como por otra parte σ es una secuencia, también es evaluable, existe f(σ) aunque les parezca raro a algunos revisores. Es posible (lo hemos comprobado) que $f(\sigma) < f(\hat{\sigma})$. Así lo indican Nagano y Moccellin (2.002) y Kalczynski afirma (comunicación particular) que si σ es una buena solución $\hat{\sigma}$ no lo es. ¿Qué condiciones debe cumplir σ para que $\hat{\sigma}$ sea tal que $f(\sigma) > f(\hat{\sigma})$?

Naturalmente no vamos a poder contestar a todas estas cuestiones, pero en los próximos trabajos exploraremos algunos aspectos relacionados.

6. REFERENCIAS.

- Brown, A. P. G.; Lomnicki, Z. A. (1966) Some applications of the 'branch-and-bound' algorithm to the machine scheduling problem. Operational Research Quarterly. **7** (4):173-86.
- Companys R. (1966) Métodos heurísticos en la resolución del problema del taller mecánic. Estudios Empresariales 66(2).
- Companys, R.; Ribas, I. (2010): New insights on the blocking flow shop problem. DIT, Working paper, (version actualizada de Companys, R,, 2009: Note on the blocking flowshop problem. DIT, Working paper, http://upcommons.upc.edu/e-prints/handle/2117/420).
- Companys, R,; Mateo, M. (2007) Different behaviour of a double branch-and-bound algorithm on $Fm|prmu|C_{max}$ and $Fm|block|C_{max}$ problems. Computers & Operations Research, **34** (4):938-53.
- Dong, X.; Huang, H.; Chen, P.(2008) An inproved NEH-based heuristic for th permutation flowshop problem, Computers & Operations Research, 35 (12) 3962-3970
- Graham R. L.; Lawler, E. L.; Lenstra, J .K.; Rinnooy Kan, A. H. G. (1979) Optimization and approximation in deterministic sequencing and scheduling: A survey. Annals of Discrete Mathematics:287-326.
- Kalczynski, P. J.; Kamburowski, J. (2008) An improved NEH heuristic to minimize makespan in permutation flow shops. Computers & Operations Research, 35(9):3001-3008.
- Leisten, R. (1990) Flowshop sequencing problems with limited buffer storage. International Journal of Production Research, **28** (11) 2085-2100
- McMahon, G. B.; Burton, P. G. (1967) Flow-shop scheduling with the branch-and-bound method. Operations Research;15(3):473-481.
- Nagano, M. S., Moccellin, J.V. (2002) A high quality constructive heuristic for flow shop sequencing'. Journal of the Operational Research Society;53:1374-1379.
- Nawaz, M.; Enscore Jr, E. E; Ham, I. (1983) A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega;11(1):91-5.

- Palmer, D. S. (1965) Sequencing jobs through a multi-stage process in the minimum total time- a quick method of obtaining a near optimum. Operational Research Quarterly. **16**:101-7.
- Pinedo, M. (1995) Scheduling: theory, Algorithms and Systems. Prentice-Hall
- Ribas, I.; Companys, R. (2009) Explotación de la reversibilidad del problema $Fm|prmu|C_{max}$ para mejorar las soluciones de las heurísticas. 3rd International Conference on Industrial Engineering ans Management, Barcelona-Terrassa, September 2-4, 2009
- Taillard, E. (1993) Benchmarks for basic scheduling problems. European Journal of Operational Research: 1/22;64(2):278-85.
- Taillard, E. (2005) Summary of best known lower and upper bounds of Taillard's instances [February 24, 2005] http://ina.eivd.ch/Collaborateurs/etd/problemes.dir/ordonnancement.dir/flowshop.dir/best_lb_up.txt
- Wang, L.; Quan-Ke-Pan; Suganthan, P. N.; Wang, W-H:; Wang, Y-M. (2010) “A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems” *Computers & Operations Research*, **37**, 509-520

ANEXO I

Para apreciar las oscilaciones producidas por el procedimiento de medida del tiempo se incluyen en la *tabla 11 bis* los valores obtenidos en otro experimento. Cabe señalar que en la *tabla 11* se pretendía excluir la operación de lectura de los datos del ejemplar y en la *tabla 11 bis* no.

		NEH0	TM1	TM1+KK	TM1+DHC
TA0001	20×5	0.000	0.000	0.000	0.000
TA0011	20×10	0.000	0.000	0.000	0.000
TA0021	20×20	0.006	0.000	0.000	0.011
TA0031	50×5	0.005	0.011	0.006	0.016
TA0041	50×10	0.006	0.011	0.011	0.038
TA0051	50×20	0.039	0.034	0.044	0.027
TA0061	100×5	0.027	0.027	0.033	0.094
TA0071	100×10	0.110	0.121	0.126	0.104
TA0081	100×20	0.126	0.131	0.137	0.417
TA0091	200×10	0.819	0.829	0.846	0.642
TA0101	200×20	1.719	1.780	0.913	2.273
TA0111	500×20	12.946	13.040	13.045	17.492
todos					

Tabla 11 bis: tiempo CPU en segundos por ejemplar al aplicar el procedimiento a los ejemplares en el supuesto $Fm / prmu / C_{max}$.

ANEXO II: Ejemplares AGUSTIN (caso Fm|prmu|C_{max})

Los resultados para los ejemplares AGUSTIN son muy coherentes con los obtenidos para las colecciones Taillard (compararar con TA0001)

n×m	NEH0	TM1	TM1+KK	TM1+DHC
15×3	0.242	0.234	0.242	0.251
15×4	0.877	0.861	0.892	0.907
15×5	1.714	1.696	1.716	1.718
15×6	2.393	2.349	2.407	2.363
16×3	0.198	0.192	0.191	0.199
16×4	0.771	0.755	0.780	0.789
16×5	1.602	1.552	1.579	1.570
16×6	2.437	2.393	2.444	2.427
17×3	0.199	0.203	0.204	0.204
17×4	0.733	0.711	0.733	0.749
17×5	1.449	1.376	1.356	1.375
17×6	2.167	2.117	2.164	2.115
todos	0.980	0.957	0.972	0.964

Tabla 13: Valores medios de la discrepancia porcentual relativa para cada procedimiento en las colecciones AGUSTIN (caso Fm/prmu/C_{max}).

n×m	NEH0	TM1	TM1+KK	TM1+DHC
15×3	6.448	6.448	6.448	7.451
15×4	7.865	7.865	7.865	7.865
15×5	8.680	9.322	8.680	9.322
15×6	9.885	9.885	9.885	9.885
16×3	4.697	4.697	4.697	4.697
16×4	9.284	8.469	8.469	8.469
16×5	8.275	9.932	8.275	9.932
16×6	9.773	9.773	9.773	9.773
17×3	5.257	5.257	5.257	5.257
17×4	7.057	7.057	6.461	6.442
17×5	12.035	12.035	12.035	12.035
17×6	9.008	9.008	9.337	9.008
todos	8.189	8.312	8.098	8.345

Tabla 14: Valores máximos de la discrepancia porcentual relativa para cada procedimiento en las colecciones AGUSTIN (caso Fm/prmu/C_{max}).

n×m	NEH0	TM1	TM1+KK	TM1+DHC
15×3	0.160	0.281	0.273	0.547
15×4	0.223	0.332	0.379	0.660
15×5	0.332	0.379	0.441	0.770
15×6	0.328	0.441	0.492	0.879
16×3	0.223	0.270	0.328	0.602
16×4	0.281	0.387	0.383	0.770
16×5	0.332	0.438	0.441	0.879
16×6	0.379	0.500	0.539	0.988
17×3	0.219	0.332	0.328	0.719
17×4	0.270	0.379	0.500	0.879
17×5	0.328	0.441	0.547	0.980
17×6	0.500	0.547	0.598	1.152

Tabla 15: Tiempos CPU en segundos por colección (1000 ejemplares, incluye lectura y escritura) para cada procedimiento en las colecciones AGUSTIN (caso $Fm/prmu/C_{max}$).

ANEXO III: Ejemplares AGUSTIN (caso Fm|block|C_{max})

Así mismo los resultados para los ejemplares AGUSTIN son muy coherentes con los obtenidos para las colecciones Taillard (compararar con TA0001)

n×m	NEH0	TM2	TM2+KK	TM2+DHC
15×3	4.326	4.445	4.600	4.345
15×4	4.716	4.819	4.934	4.729
15×5	4.533	4.641	4.734	4.555
15×6	4.475	4.563	4.603	4.511
16×3	3.804	3.975	4.128	3.859
16×4	4.126	4.262	4.392	4.179
16×5	4.114	4.211	4.326	4.159
16×6	3.914	4.045	4.101	3.941
17×3	3.917	4.049	4.236	3.961
17×4	4.271	4.404	4.557	4.343
17×5	4.227	4.332	4.402	4.264
17×6	3.939	4.063	4.130	3.974
todos	4.197	4.317	4.429	4.235

Tabla 13: Valores medios de la discrepancia porcentual relativa para cada procedimiento en las colecciones AGUSTIN (caso Fm/block/C_{max}).

n×m	NEH0	TM2	TM2+KK	TM2+DHC
15×3	11.568	12.615	12.615	11.568
15×4	13.908	13.908	13.908	13.908
15×5	13.212	12.352	12.807	12.166
15×6	10.983	10.983	11.436	10.983
16×3	11.606	12.176	12.176	11.606
16×4	11.902	11.902	11.923	11.902
16×5	12.553	12.553	12.553	12.553
16×6	11.101	11.802	11.802	11.101
17×3	13.085	13.085	15.162	13.085
17×4	10.267	11.395	12.612	11.648
17×5	10.675	10.675	10.945	10.945
17×6	10.757	10.757	10.757	10.757
todos	11.801	12.017	12.391	11.852

Tabla 14: Valores máximos de la discrepancia porcentual relativa para cada procedimiento en las colecciones AGUSTIN (caso Fm/block/C_{max}).

n×m	NEH0	TM2	TM2+KK	TM2+DHC
15×3	0.332	0.219	0.488	0.879
15×4	0.500	0.328	0.328	0.613
15×5	0.277	0.379	0.770	0.707
15×6	0.332	0.438	0.551	0.820
16×3	0.391	0.492	0.551	0.500
16×4	0.270	0.660	0.379	0.609
16×5	0.609	0.441	0.820	0.770
16×6	0.328	0.488	0.492	0.930
17×3	0.438	0.328	0.598	0.609
17×4	0.332	0.719	0.441	0.719
17×5	0.660	0.488	0.934	0.941
17×6	0.820	0.602	0.551	1.051

Tabla 15: Tiempos CPU en segundos por colección (1000 ejemplares, incluye lectura y escritura) para cada procedimiento en las colecciones AGUSTIN (caso $Fm \,/\, block \,/ C_{max}$).

ANEXO IV: Desempate en el primer paso mediante S_4

Hemos utilizado para desempatar en el orden LPT una regla basada en el algoritmo de los trapecios (Companys, 1966). La regla es la siguiente:

Para cada pieza i se define un índice $S_{4,i}$ = max $\{S_{1,i} ; S_{2,i}\}$donde

$$S_{1,I} = \sum_{j=1}^{m} (m - j) \cdot p_{j,i} \quad y \quad S_{2,I} = \sum_{j=1}^{m} (j - 1) \cdot p_{j,i}$$

(conviene recordar que ordenado por $S_{3,I} = S_{1,i} - S_{2,I}$ no decreciente se obtiene una permutación acorde a la heurística de Palmer, 1965). Si dos piezas h e i son tales que $P_h = P_i$ se otorga prioridad en el orden inicial a aquella de mayor valor S_4.

desempate	**paso 1**	**NO**	**NO**	**SI**	**SI**
desempate	**paso 2**	**NO**	**SI**	**NO**	**SI**
TA0001	20×5	2.492	2.130	2.591	**2.105**
TA0011	20×10	4.174	**4.107**	4.175	**4.107**
TA0021	20×20	**3.360**	3.495	3.437	3.585
TA0031	50×5	0.581	0.541	0.610	**0.435**
TA0041	50×10	4.973	4.387	4.714	**4.222**
TA0051	50×20	5.861	**5.671**	5.893	5.831
TA0061	100×5	0.378	0.381	**0.342**	0.385
TA0071	100×10	2.016	1.768	1.876	**1.602**
TA0081	100×20	5.196	**4.904**	5.011	5.074
TA0091	200×10	1.148	1.053	1.205	**0.957**
TA0101	200×20	4.196	**3.849**	4.030	3.909
TA0111	500×20	1.985	**1.819**	1.929	1.989
todos		3.030	**2.842**	2.984	2.850

Tabla 16: *Comparación de los resultados de aplicación del algoritmo NEH a los ejemplares de Taillard (caso $Fm / prmu / C_{max}$) sin y con reglas adicionales de desempate en el primer paso (regla S_4) y en el segundo (regla TM1)*

desempate	paso 1	NO	NO	SI	SI
desempate	**paso 2**	**NO**	**SI**	**NO**	**SI**
TA0001	20×5	4.894	5.164	**4.817**	4.934
TA0011	20×10	**5.219**	5.264	**5.219**	5.264
TA0021	20×20	**3.365**	3.386	3.373	3.431
TA0031	50×5	**7.746**	7.951	7.752	7.800
TA0041	50×10	7.522	7.295	7.378	**7.272**
TA0051	50×20	6.846	7.136	**6.799**	7.166
TA0061	100×5	7.924	7.855	**7.706**	7.935
TA0071	100×10	7.517	7.273	7.393	**7.209**
TA0081	100×20	5.516	5.662	**5.289**	5.573
TA0091	200×10	7.608	**7.310**	7.429	7.338
TA0101	200×20	5.244	**5.052**	5.066	5.228
TA0111	500×20	4.317	4.296	4.436	**4.263**
todos		6.143	6.137	**6.055**	6.118

Tabla 17: *Comparación de los resultados de aplicación del algoritmo NEH a los ejemplares de Taillard (caso Fm /block /C_{max}) sin y con reglas adicionales de desempate en el primer paso (regla S_4) y en el segundo (regla TM2)*

El procedimiento de desempate es efectivo en el caso Fm|block|C_{max}, pero de efectos dudosos en el caso Fm|prmu|C_{max}.